

CLAIMS

What is claimed is:

1. A method to determine one or more shortest paths through a portion of a computer network from a source vertex to one or more destination vertices according to a link-state protocol, comprising:

processing a graph representation of the network portion, the graph representation including nodes and edges representing the vertices and connections therebetween, respectively

wherein

the processing includes operating on the graph representation according to a Dijkstra-like algorithm,

a subset of the Dijkstra-like algorithm processing includes candidate list processing, to maintain and operate upon a candidate list of nodes that have been visited in the Dijkstra-like algorithm processing; and

the candidate list processing is optimized relative to standard Dijkstra algorithm processing for the link-state protocol.

2. The method of claim 1, wherein the candidate list processing includes:

maintaining the candidate list of nodes as a Fibonacci heap of Fibonacci nodes, wherein the Fibonacci heap of Fibonacci nodes is stored in a generic format that is independent of the link-state protocol; and

operating on the Fibonacci heap of Fibonacci nodes to determine one or more shortest paths by processing the Fibonacci heap of Fibonacci nodes according to a particular algorithm that is independent of the link-state protocol.

3. The method of claim 2, wherein the candidate list processing includes initially accepting a prior-created generic description of the Fibonacci heap of Fibonacci nodes that is independent of the particular link-state protocol.

4. The method of claim 3, wherein initially accepting a prior-created generic description for the Fibonacci heap of Fibonacci nodes includes accepting the prior-created generic description via a generalized application programming interface.

5. The method of claim 2, wherein the step of processing the Fibonacci heap of Fibonacci nodes includes considering the Fibonacci nodes to determine which nodes to include in the one or more shortest paths.

6. The method of claim 2, wherein the step of processing the Fibonacci heap of Fibonacci nodes includes a relax-key operation performed in an iterative manner.

7. The method of claim 2, including:
initially allocating memory for a list of pointers to the Fibonacci nodes,
wherein the step of operating on the Fibonacci nodes includes maintaining the list of pointers to the Fibonacci nodes.

8. The method of claim 2,
wherein the step of operating on the Fibonacci nodes includes accessing a comparison function associated with the Fibonacci heap in a manner that is independent of the particular link-state protocol.

9. The method of claim 1, wherein:
the link-state protocol is a link-state routing protocol.